



## Studying Transfer of Learning using a Brain-Inspired Spiking Neural Network in the Context of Learning a New Programming Language

Fard, M., Petrova, K., Kasabov, N., & Wang, G. (Accepted/In press). Studying Transfer of Learning using a Brain-Inspired Spiking Neural Network in the Context of Learning a New Programming Language. In *Proceedings of the 8th IEEE conference on CSDE, Brisbane, 8-10.12.2021* (pp. 1). (IEEE Conference on Computer Science, Data Science and Engineering ). IEEE. <https://ieeecsde.org/2021/>

[Link to publication record in Ulster University Research Portal](#)

### Published in:

Proceedings of the 8th IEEE conference on CSDE, Brisbane, 8-10.12.2021

### Publication Status:

Accepted/In press: 30/11/2021

### Document Version

Author Accepted version

### General rights

Copyright for the publications made accessible via Ulster University's Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

### Take down policy

The Research Portal is Ulster University's institutional repository that provides access to Ulster's research outputs. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact [pure-support@ulster.ac.uk](mailto:pure-support@ulster.ac.uk).

# Studying Transfer of Learning using a Brain-Inspired Spiking Neural Network in the Context of Learning a New Programming Language

Mojgan Hafezi Fard

*School of Engineering, Computer and Mathematical Sciences  
Auckland University of Technology  
Auckland, New Zealand  
mojgan.hafezi.fard@aut.ac.nz*

Nikola Kasabov

*School of Engineering, Computer and Mathematical Sciences  
Auckland University of Technology  
Auckland, New Zealand  
Intelligent Systems Research Centre, Ulster University, UK  
nkasabov@aut.ac.nz*

Krassie Petrova

*School of Engineering, Computer and Mathematical Sciences  
Auckland University of Technology  
Auckland, New Zealand  
krassie.petrova@aut.ac.nz*

Grace Y. Wang

*Department of Psychology  
Auckland University of Technology  
Auckland, New Zealand  
grace.wang@aut.ac.nz*

**Abstract**— Transfer of learning (TL) has been an important research area for scholars, educators, and cognitive psychologists for over a century. However, it is not yet understood why applying existing knowledge and skills in a new context does not always follow expectations, and how to facilitate the activation of prior knowledge to enable TL. This research uses cognitive load theory (CLT) and a neuroscience approach in order to investigate the relationship between cognitive load and prior knowledge in the context of learning a new programming language. According to CLT, reducing cognitive load improves memory performance and may lead to better retention and transfer performance. A number of different frequency-based features of EEG data may be used for measuring cognitive load. This study focuses on analysing spatio-temporal brain data (STBD) gathered experimentally using an EEG device. An SNN based computational architecture, NeuCube, was used to create a brain-like computation model and visualise the neural connectivity and spike activity patterns formed when an individual is learning a new programming language. The results indicate that cognitive load and the associated Theta and Alpha band frequencies can be used as a measure of the TL process and, more specifically, that the neuronal connectivity and spike activity patterns visualised in the NeuCube model can be interpreted with reference to the brain activities associated with the TL process.

**Keywords**— *transfer of learning, cognitive load, learning computer programming, spiking neural networks, SNN, NeuCube, EEG*

## I. INTRODUCTION

The fundamental goal of education and learning systems is to enable a learner to apply the learned material to different contexts and extend one's learning to new situations [1]. However, the desired transfer of knowledge and learned skills from one context to another does not always happen smoothly or quickly [2].

Transfer of learning (TL) has been an important research area for scholars, educators, and cognitive psychologists for over a century. Various factors such as classroom environment, attention-getting techniques, and mental manipulations have been considered and investigated regarding how they may influence the transformation of sensory information into long-term memory and conceptual understanding [3].

A specific area of interest is studying how an understanding of brain activities during the process of TL can help improve learning in a given context, for example, learning a computer programming language. As teaching and learning how to program has been found difficult by both learners and educators, the topics of how to improve student learning and support novice programmers have attracted a substantial body of research [4].

It is assumed that the way students learn programming and build new knowledge may be influenced by their background and existing skills [5]. However, students in introductory programming courses may have different prior knowledge in the area of programming. Educators need to consider this when formulating learning goals and preparing teaching materials to facilitate the TL process of the different learner groups more effectively. This is even more important in the now becoming widely adopted online education as online education models tend to increase learner's cognitive load [6].

Research attempting to understand how the TL process works by gathering data through methods such as questionnaires, surveys and interviews has not yet explained how TL occurs in the human brain [3]. A potential reason for this may be the lack of accuracy in the data collected using these methods. In self-reported subjective data, bias may be introduced by factors such as personal preferences and cultural background [7].

In recent years, there has been an increased interest in studying how to activate prior knowledge stored in the human brain - as the brain is the seat of learning [8]. Learning impacts the brain by changing the connections among neurons which leads to changes in the existing internal synaptic structures [9]. This fact has prompted scientists to take a neuroscience approach towards studying TL.

## II. A NEUROSCIENCE APPROACH TO THE STUDY OF TRANSFER OF LEARNING

Connecting neuroscience to education to study the TL process of learners taking challenging subjects such as computer programming may lay the scientific groundwork and build a strong research foundation for a better understanding of learning and its processes [10]. Knowing more about the neural mechanisms activated when a

programmer is at work may help find better approaches to programmer training [11]. Recent advances in brain imaging techniques have provided additional opportunities for researchers to collect spatio-temporal brain data (STBD) and study in more depth the activities of the human brain that occur during learning a computer programming language [12]. However, our knowledge about the patterns of neural activity related to learning computer programming or performing a programming task is still very limited [11].

#### A. Cognitive Load Theory and Transfer of Learning

Cognitive load is a concept related to the use of the human brain's working memory (WM) which is limited in its capacity. WM can be defined as the capability of the individual to "simultaneously manipulate some information while maintaining other information" [13]. Thus, cognitive load signifies the total amount of WM resources used while a cognitive task is performed [14]. The study of factors that impact positively on WM performance and thus may improve learner performance and task completion has already attracted significant research interest [13].

Cognitive load theory (CLT) was developed by John Sweller to describe the relationship between WM and the cognitive effort used during a learning task [15]. According to CLT, WM usage and learner performance are inversely related. A low level of WM usage (i.e., low cognitive load) indicates that the learner has relevant prior knowledge and/or expertise about the learning task. In contrast, high WM usage (i.e., high cognitive load) indicates a lack of specific prior knowledge or expertise in relation to the task [7]. Therefore, a reduction in cognitive load improves learner outcomes (i.e., better retention and transfer performance) [16].

Cognitive load can be used to predict learner performance. For example, it was demonstrated that measuring cognitive load during a computer programming comprehension exercise is a more reliable way to judge programmers' expertise compared to evaluating the correctness and/or the speed of completing the task [17]. As relevant prior knowledge and/or expertise are expected to have a positive impact on learner performance, it may be possible to use cognitive load as a measure of the TL process through which the learner applies relevant prior knowledge /expertise to a new task.

Based on the above, we formulated the following research question: What is the relationship between prior knowledge and cognitive load in the context of learning a new programming language? We hypothesised that prior knowledge of computer programming would reduce cognitive load, thus leading to a better learning outcome and transfer performance.

Electroencephalogram (EEG) data can provide evidence about WM performance; therefore, EEG data can be used to quantify cognitive workload [18]. According to [19], EEG can also provide data for the modelling learners' mental state. The ratios between different frequency-based EEG data features are of specific significance for measuring cognitive load [20]-[21]. Notably, it was found that the ratio between the Alpha power band (8-12 Hz) and the Theta power band (4-8 Hz) was particularly important [20]. The Alpha and Theta bands mainly reflect cognitive and memory performance; for example, an increase in Alpha and an accompanying decrease in Theta indicate 'good' WM performance (reduced cognitive load) [22].

#### B. Spiking Neural Networks for the Study of Transfer of Learning based on EEG Data

Spiking Neural Networks (SNNs) have proved to be one of the most successful approaches towards modelling the behaviour and learning potential of the brain [23]. SNNs are especially efficient in analysing STBD collected by various imaging techniques, including EEG and fMRI [24]-[25]. SNNs are more biologically realistic than other artificial neural networks (ANNs) as they mimic more closely neurons' biological activity, using discrete spikes to compute and transmit information dynamically. The computational method used in SNNs is inspired by natural neural network activities that are captured in STBD. Spikes across spiking neurons can encode both spatial and temporal information. SNNs process STBD after the data is encoded into spike trains (spike information processing) [24].

Previous studies have produced promising results on analysing STBD by using NeuCube - a unique computational architecture built on an SNN framework which allows to create brain-like computational models [26]. The architecture of NeuCube was introduced in [24] and then realised in various software implementations for the creation of application systems [27], [30]. To answer our research question, we use NeuCube to process STBD related to learning a computer programming language, and deploy NeuCube's visualization functionality in order to observe and interpret the neural connections and connectivity patterns formed while processing STBD input.

### III. MATERIALS AND METHODS

The experiment carried out in this study involved exploring similarities and differences between brain activities of research participants using EEG as a data collection technique.

#### A. Learning Tasks

The design of the learning tasks was inspired in part by the work presented in [31]-[32]. Research participants were asked to complete 17 learning tasks: each task comprised a computer programming comprehension exercise followed by a four-answer multi-choice question. The programming exercises were written for the C programming language. The learning tasks were arranged in order of increasing difficulty and were displayed on a laptop monitor, in the same order for all participants.

The experiment starts by displaying an instruction page and an OK button. Clicking the OK button takes the participant to the first programming exercise. In order to prevent any unnecessary stress, a time limit was not set for answering each question. After reading the multi-choice question, the participant needs to select one of the answers, using a mouse click. After that, the participant is shown a screen displaying the word 'Relax'. The next task is displayed automatically after two seconds. The process is repeated for all learning tasks.

#### B. Research Participants

Ethical approval was obtained from the authors' institutional ethics committee. Potential participants were invited to volunteer through email invitations sent to all first-year computer science students enrolled in a Bachelor's program at the authors' university. Participants who expressed interest were asked to complete a screening questionnaire, providing information about the number of

programming languages they already knew, their self-estimated level of knowledge, and age and gender.

All eight recruited participants were male, between 18 and 21 years old. All of them were about to start learning the C programming language in their introductory programming course. All knew at least one programming language (different from C), with the level of prior knowledge varying from beginner to average.

### C. Data Collection Method

The OpenSesame software (<https://osdoc.cogsci.nl/>) was used to build and present the research participants' learning tasks, interfacing it with the EEG device. The EEG device used for gathering participant brain activity data was Emotive Epoch X (<https://www.emotiv.com/epoc-x/>). It utilizes 14 channels: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4. The device was chosen as it was explicitly designed for scalable and contextual human brain research. In particular, the device is easily moved and easy to use, with a minimised set-up time). Even more importantly, the quality of the output is high (professional grade' brain data).

Relevant event markers were configured in both the EEG device software system and the presentation software. The headset electrodes were placed on participants' scalps following the 10-20 international system to record the brain's electrical activity resulting from the brain cells' neuronal firing. The sampling rate was configured as 256 Hz.

## IV. COMPUTATIONAL MODELLING IN NEUCUBE AND ANALYSIS OF B RESULTS

At the pre-processing stage, the experimental dataset was reorganised in order to meet NeuCube input requirements. The data analysis process comprised two stages: first, the data were analysed in order to determine which parts of the participant's brain were more activated during the programming tasks, by identifying the most active channels. Subsequently, the data were analysed to identify the relationships between the wavebands occurring during the programming task and, more specifically, finding the relationship between the Alpha and Theta frequencies. The results of the two analyses are discussed further below.

### A. Data Pre-Processing

The output produced by the EEG device was a file of raw data in EDF format which for further processing, it was converted to a CSV data file. Next, the basic filtering and individual component analysis (ICA) from MATLAB's EEGLAB toolbox were performed to remove noise from the data. These two techniques are commonly used to reduce the noise level without losing meaningful information from the dataset.

Considering the study's research question, the wavebands were particularly important features for analysis; therefore the entire spectrum of frequencies was needed to be acquired from the recorded signals. However, those parts of the participant's brain that were most active during the programming tasks performance was needed to be identified before this process. To accomplish this, NeuCube was used to build a 3D brain-like model to visualise the neural interactions and the spike connectivity among the EEG channels.

As described earlier, the dataset in this study was collected from eight individuals while they were completing 17 learning tasks related to computer programming. Two event

markers were created for each task (trial), indicating the start and the end of the task (the end of the task was signalled when the participant clicked the selected answer to the multi-choice question). The event markers were saved in the CSV dataset file as an individual column.

To feed the dataset into the NeuCube, the dataset was organised as individual spatio-temporal samples. 17 independent trials of 3.5 seconds duration were extracted from each original dataset belonging to each participant. The second marker index was used as the time point to separate the 17 trials as individual samples. Each sample consisted of 1250 rows of ordered times (temporal features) and 14 columns of spatial features corresponding to 14 EEG channels.

### B. The First Stage of the Analysis

A NeuCube model comprises an encoding module, a 3D SNN cube (SNNc), an eSNN classifier, and an optimization module [26]. The SNNc is structured according to a brain template (in this case, the Talairach template) with 1471 spiking neurons; each represents a small area from the template and the brain. This allows for a trained SNNc to be interpreted in terms of functional connectivity related to the brain areas that are measured in STBD (e.g., an EEG dataset).

For the analysis, participants were divided into two groups based on their self-reported level of prior programming knowledge. Those at level average and above were labeled as 'with sufficient prior knowledge' (SPK), while participants at level beginner or below average were labeled as 'insufficient prior knowledge' (IPK).

Each individual's participant dataset was loaded into the input encoding module of NeuCube. At this step, the continuous data stream is converted into discrete spike trains using a threshold base encoding algorithm suitable for processing in an SNN cube [27]. The model is initialised by mapping the input variables onto the spatially located spike neurons in the SNNc.

Next, the model is trained on the data in the SNNc, deploying in a two-stage learning process. At the first stage (unsupervised learning), the data's hidden (spatio-)temporal relationships were encoded as neuronal connection weights; this results in SNNc learning (spatio-)temporal relations from input data. SNNc is trained by using the spike-timing-dependent synaptic plasticity (STDP) learning rule. According to the rule, if neuron  $j$  fires before neuron  $i$ , the weight of the connection from neuron  $j$  to neuron  $i$  will increase; if the order reverses, the weight of the connection from neuron  $i$  to neuron  $j$  will decrease [29]. The STDP rule extends the Hebbian learning rule, according to which the constant interaction between two neurons strengthens their connection [33].

The second training stage is supervised learning which trains the model with regard to the target output class associated with each (spatio-)temporal sample. As the aim was to identify the patterns underlying neuronal connectivity and spike activity for each individual participant's dataset, the supervised training stage was not relevant to the analysis.

NeuCube provides different techniques for network analysis, such as clustering according to spike activity or connection weight, and graphical representation of cluster interaction showing information exchange [30]. After unsupervised training, we applied clustering according to the spiking activity method for all 14 input features and then

extracted an average input interaction to visualise the interaction between features.

### C. Results of the First Stage of the Analysis

The diagrams in Fig.1 represent one-to-one interaction between the input neurons and their surrounding clusters based on average spike exchange over the time of duration of the EEG signals. In each figure, 14 input neuron clusters belong to the 14 features (channels) of the EEG device. The lines represent the interaction between features. Thicker lines demonstrate more interaction between the features. In Fig.1, the lines representing the interaction between the cluster neuron of channels T7 and F7 of the EEG device showed the highest amount of interaction for both groups (SPK) and (IPK). Channel T7 is located in the temporal lobe of the left hemisphere of the brain, approximately at the ear level of the skull. This part of the brain is the first area responsible for interpreting the information in the form of sounds from the ear. Channel F7 is located in the frontal lobe of the left hemisphere of the brain. This part of the brain also creates and controls spoken and written language output and is responsible for attention gaining, WM memory and decision making. Considering the primary functionalities of these two brain areas and the purpose of this study which is related to prior learning, WM and programming languages, we selected channel F7 to further test our hypothesis.

### D. The Second Stage of the Analysis

At this stage of analysis, the power spectrum of channel F7 was obtained using the Fast Fourier Transform (FFT) method (in MATLAB). The frequency bands extracted by this method are commonly used to study different mental states and learning processes [34]. The six frequency bands extracted in this stage are known as; Delta (0.5-4Hz), Alpha(8-12Hz), Theta (4-8 Hz), Beta High (16-25), Beta Low (12-16) and Gamma (25-45 Hz).

NeuCube was used to investigate the relationships between these features/frequency bands. Seventeen samples were extracted from each one of the datasets. Each sample consisted of 125 rows of ordered time (temporal) features and six columns of waveband features (Delta, Alpha, Theta, Beta H, Beta L, Gamma). The NeuCube encoding, initialization, and unsupervised training processes were deployed in the same way as at the first stage of the analysis. However, after clustering the input data based on the spike communication, we used the neuronal proportion graph to represent the relative percentage of neurons out of the total number of neurons in the SNNc. Neuronal proportion represents the strength of neuronal interaction between the wavebands in terms of the percentage of neurons in the cube that belongs to an input neuron cluster.

### E. Results of the Second Stage of the Analysis

The diagrams in Fig. 2 represent the percentage of six features (wavebands) in the trained SNNc out of the total number of neurons in the cube. For each participant, we compared the percentage of neurons between the features, specifically Theta and Alpha (which are the waves of interest in this study), to find information regarding the relationship between them. As a result, for participants labeled as SPK, the neuron proportion related to the Alpha band is higher than the neuron proportion for the Theta band, while for the participants labeled as IPK, the neuron proportion for the Theta band is higher when compared to the Alpha band (Fig. 2). For the ease of comparison for each individual and

subsequently, as a group, the bar graph (Fig. 3) was created which displays the neuron proportion in percentage (vertical axis) related to the Alpha and Theta band features for each of the eight participants (horizontal axis). The first set of four bar pairs represent participants of type IPK, while the second set of four bar pairs represent the SPK participants.

## V. DISCUSSION

Overall, the findings of this research demonstrated that modelling and analysing the STBD related to TL by using a brain-inspired SNN approach, specifically NeuCube, can reveal meaningful brain activity patterns manifesting the relationship between prior knowledge and cognitive load. The study results confirm the hypothesis that prior knowledge of computer programming reduces cognitive load, thus leading to a better learning outcome and transfer performance.

The results from the cluster analysis based on average input interaction demonstrated a broad interaction and spike activity among neurons in the frontal lobe (6 channels located in the frontal lobe) and temporal lobe of the brain (T7, T8). However, the highest neuronal interaction among all participants were registered at F7 and T7, both located in the left hemisphere, frontal and temporal lobe respectively.

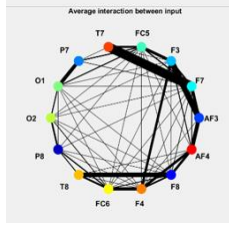
Considering the above results, we can conclude that the TL during performing a programming task favours the left hemisphere of the brain in the frontal and temporal lobes. This finding is in line with a recent study using fMRI scan, where the left hemisphere of the brain is found to correspond with solving the programming tasks. More interestingly, this area also correlates with language processing [11].

In the second stage of the analysis, results from neuronal interaction based on the neuron proportion metric of NeuCube showed an increase in the Theta band feature but a decrease in the Alpha band feature for the dataset of IPK participants (Fig.2). In contrast, the same analysis for the dataset of the SPK participants showed a decrease in the Theta and an increase in the Alpha band feature.

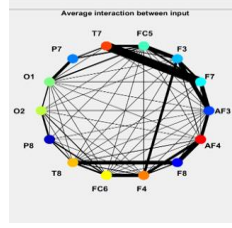
We can conclude that the insufficiency in prior knowledge was demonstrated by an increase in Theta and a decrease in Alpha, while having sufficient prior knowledge in programming can decrease Theta and increase Alpha band. These results also indicate that the cognitive load of the IPK participants was high, indicating a poor learning outcome and low transfer performance due to the decreased WM efficiency. Indeed, these participants' scores were poor and their prior knowledge of programming was insufficient. The self-reported information of mentioned participants also was in line with the result of the experiment which confirmed insufficiency of their prior knowledge.

The results for the SPK participants indicate a low cognitive load that allows WM to work with greater efficiency, potentially leading to better learning outcomes and transfer performance. Indeed, the data in these datasets were gathered from participants with sufficient prior knowledge as their self-reported result confirmed the higher level of knowledge compared to participants from another group (IPK). The scores of these participants were higher than another group in terms of correct answers.

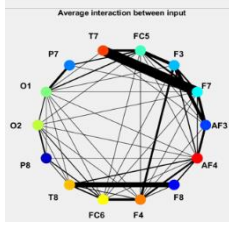
The findings of the reverse correlation of these bands are aligned with results reported in previous studies about the relationships between the Theta and Alpha wavebands with regards to cognitive load and performance [22], [18], [16].



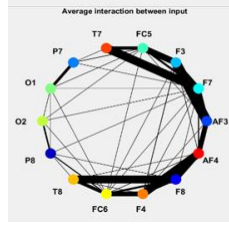
Dataset 26 (SPK)



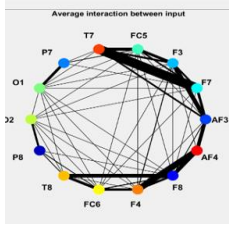
Dataset 22 (IPK)



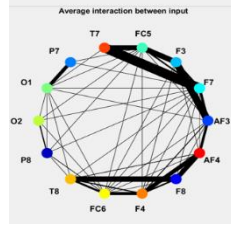
Dataset 24 (SPK)



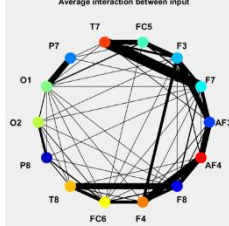
Dataset 18 (IPK)



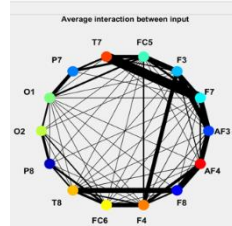
Dataset 16 (SPK)



Dataset 12 (IPK)



Dataset 14 (SPK)

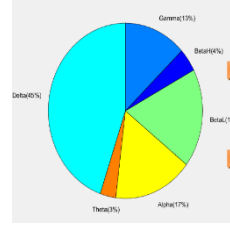


Dataset 10 (IPK)

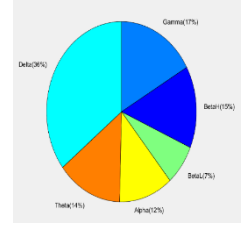
Fig. 1. **SNNc visualisation:** Average input interaction in terms of spike communication between clusters of input neurons which correspond to the 14 channels used for EEG data collection. SPK: sufficient prior knowledge; IPK: insufficient prior knowledge.

## VI. CONCLUSION

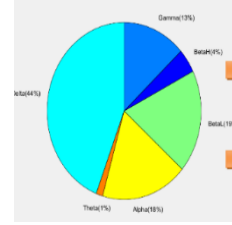
To explore the relationship between prior knowledge and cognitive load, this research used STBD and the brain-inspired SNN computational architecture of NeuCube to study the neuronal connectivity and spike activities underlying the brain activities occurring when individuals learn a new programming language. Inferences about cognitive load were drawn from data about the change patterns in the Alpha and Theta frequencies. During the experiment, participants with sufficient prior knowledge of computer programming performed better in their learning tasks compared to participants with insufficient prior knowledge. The analyses of the outputs of the SNN model demonstrated that during the experiment, the cognitive load of participants with prior knowledge of computing was reduced compared to the cognitive load of participants with insufficient prior knowledge.



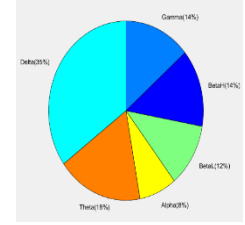
Dataset 26 (SPK)



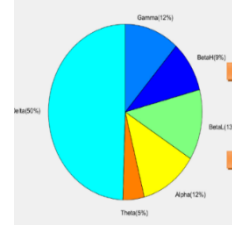
Dataset 22 (IPK)



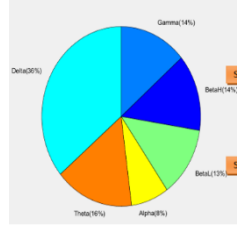
Dataset 24 (SPK)



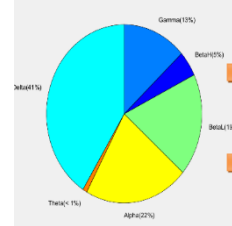
Dataset 18 (IPK)



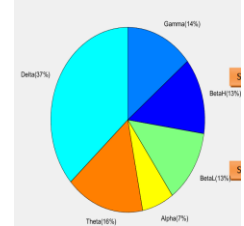
Dataset 16 (SPK)



Dataset 12 (IPK)



Dataset 14 (SPK)



Dataset 10 (IPK)

Fig. 2. **SNNc visualisation:** Neuron proportion in terms of spike communication between clusters of input neurons which correspond to six frequency bands. SPK: sufficient prior knowledge; IPK: insufficient prior knowledge.

Even though the participant sample EEG data was small, the results allow us to conclude that cognitive load and the associated Theta and Alpha frequencies can be used as a measure of the TL process. More specifically, it was demonstrated that the neuronal connectivity and spike activity patterns visualised in the NeuCube model can be used to understand and interpret brain activities occurring during the TL process. Further research may include the investigation of the effect of learners' prior knowledge on cognitive load considering other EEG channels such as T7 (located in the temporal lobe) and the identification of other patterns of change in the Theta and Alpha frequencies.

## ACKNOWLEDGMENT

Multiple iterations of the NeuCube architecture have been developed under the supervision of Professor N. Kasabov. NeuCube is freely available for study and research at [www.kedri.aut.ac.nz/neucube](http://www.kedri.aut.ac.nz/neucube) and also at [www.neucube.io](http://www.neucube.io) (also funded by Auckland University of Technology).



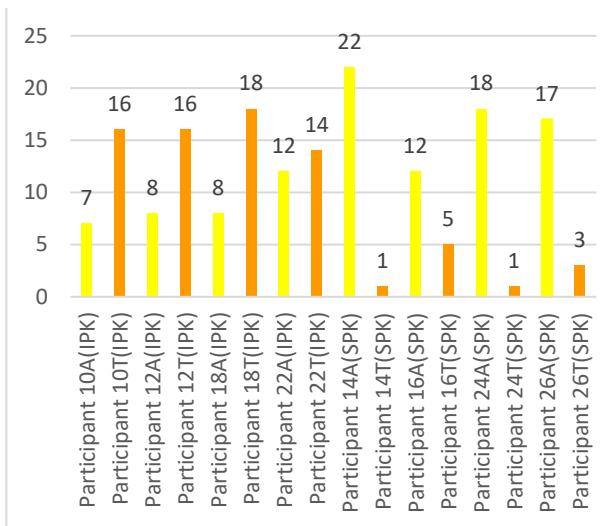


Fig. 3. Comparison of Alpha (A) and Theta (T) features in terms of neuronal proportion (in percentage), for all participants (SPK and IPK).

A team including researchers from Auckland University of Technology, Dr. Enmei Tu from Shanghai Jiao Tong University and the team of Prof. Hou from the CASIA Beijing developed the MATLAB version of NeuCube.

#### REFERENCES

- [1] C. Bossard, G. Kermarrec, C. Buche and J. Tisseau, "Transfer of learning in virtual environments: a new challenge?," *Virtual Reality*, vol. 12, no. 3, pp. 151-161, 2008.
- [2] D.N. Perkins and G. Salomon, "Transfer of learning. International encyclopedia of education," Ed. Torsten Husten, vol. 2, 1992.
- [3] S. H. Forrester, "Transfer of learning and music understanding: A review of literature", Update: Applications of Research in Music Educ., vol.37, no. 1, pp. 30-35, 2018.
- [4] A. Decker, "SIGCSE top ten symposium papers of all time award winners," *ACM SIGCSE Bulletin*, vol. 51, no. 2, pp. 14-15, 2019.
- [5] B. A. Simonsmeier, M. Flaig, A. Deiglmayr, L. Schalk, and M. Schneider, "Domain-specific prior knowledge and learning: A meta-analysis," *Educ. Psychol.*, pp. 1-24, 2021.
- [6] S. Gregory and M. Bannister-Tyrrell, "Digital learner presence and online teaching tools: higher cognitive requirements of online learners for effective learning," *Res. and Pract. in Technol. Enhanced Learn.*, vol. 12, no. 1, pp. 1-17, 2017.
- [7] Crk, T. Kluthe and A. Stefik, "Understanding programming expertise: An empirical study of phasic brain wave changes", *ACM Trans. Comput. Human Interaction*, vol. 23, no. 1, pp. 1-29, 2015.
- [8] G. Brod, M. Werkle-Bergner, and Y. L. Shing, "The influence of prior knowledge on memory: a developmental cognitive neuroscience perspective," *Frontiers in behave. neuroscience*, vol. 7, p. 139, 2013.
- [9] P. Moreno- Aparicio and A. Rodríguez-Moreno, "The Impact of Studying Brain Plasticity," *Frontiers in Cellular Neuroscience*, article 2019.
- [10] P. Giannopoulou, M.-A. Papalaskari, and S. Doukakis, "Neuroeducation and Computer Programming: A Review," *GeNeDis* 2018, pp. 59-66.
- [11] J. Rosen, "This is your brain on code: JHU researchers decipher neural mechanics of computer programming", *The Hub*, Dec 8, 2021. Accessed on: 25- Sep- 2021. [Online]. Available: <https://hub.jhu.edu/2020/12/17/brain-activity-while-reading-code/>.
- [12] S. Doukakis, "Exploring brain activity and transforming knowledge in visual and textual programming using neuroeducation approaches", *AIMS Neuroscience*, vol. 6, no. 3, pp. 175-190, 2019. Available: 10.3934/neuroscience.2019.3.175.
- [13] Y R. Blasiman and C. Was, "Why is working memory performance unstable," *A review of 21 factors. Europe's journal of psychol.*, vol. 14, no.1, pp. 188-231, 2018.
- [14] Ø. Anmarkrud, A. Andresen, and I. Bråten, "Cognitive load and working memory in multimedia learning: Conceptual and measurement issues," *Educ. Psychol.*, vol. 54, no. 2, pp. 61-83, 2019.
- [15] E. W. Anderson, K. C. Potter, L. E. Matzen, J. F. Shepherd, G. A. Preston, and C. T. Silva, "A user study of visualization effectiveness using EEG and cognitive load," in *Computer graphics forum*, 2011, vol. 30, no. 3, pp. 791-800: Wiley Online Library
- [16] H. Xie et al., "The more total cognitive load is reduced by cues, the better retention and transfer of multimedia learning: A meta-analysis and two meta-regression analyses," *PloS one*, vol. 12, no. 8, 2017.
- [17] L. Gonçalves, K. Farias, B. da Silva, and J. Fessler, "Measuring the cognitive load of software developers: a systematic mapping study," in *2019 IEEE/ACM 27th Int. Conf. on Prog. Comprehension (ICPC)*, 2019, pp. 42-52: IEEE.
- [18] S. Puma, N. Matton, P.-V. Paubel, É. Raufaste, and R. El-Yagoubi, "Using theta and alpha band power to assess cognitive workload in multitasking environments," *Int. J. of Psychol.*, vol. 123, pp. 111-120, 2018.
- [19] C. Mills, I. Fridman, W. Soussou, D. Waghray, A. M. Olney, and S. K. D'Mello, "Put your thinking cap on: detecting cognitive load using EEG during learning," in *Proceedings of the seventh int. learning analytics & knowledge conf.*, 2017, pp. 80-89: ACM.
- [20] N. Friedman, T. Fekete, Y. a. K. Gal, and O. Shriki, "EEG-based Prediction of Cognitive Load in Intelligence Tests," *Frontiers in Human Neuroscience*, vol. 13, p. 191, 2019.
- [21] C. D. Conrad and M. Bliemel, "Psychophysiological measures of cognitive absorption and cognitive load in e-learning applications," 2016.
- [22] W. Klimesch, "EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis," *Brain Res. Rev.*, vol. 29, no. 2-3, pp. 169-195, 1999.
- [23] W. Maass, "Networks of spiking neurons: The third generation of neural network models", *Neural Networks*, vol. 10.9, 1997, 1657-1671, [https://doi.org/10.1016/S0893-6080\(97\)00011-7](https://doi.org/10.1016/S0893-6080(97)00011-7)
- [24] N. Kasabov, "NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data", *Neural Netw.*, vol. 52, pp. 62-76, 2014.
- [25] L. Lobo, J. Del Ser, A. Bifet and N. Kasabov, "Spiking neural networks and online learning: An overview and perspectives," *Neural Netw.*, vol. 121, pp. 88-100, 2020.
- [26] M. H. Fard, K. Petrova, M. Doborjeh, and N. Kasabov, "Using EEG Data and NeuCube for the Study of Transfer of Learning," in *2020 Int.*
- [27] N. Kasabov et al., "Evolving spatio-temporal data machines based on the NeuCube neuromorphic framework: Design methodology and selected applications," *Neural Networks*, vol. 78, pp. 1-14, 2016.
- [28] N. Kasabov, L. Zhou, M. Gholami Doborjeh, J. Yang, "New algorithms for encoding, learning and classification of fMRI data in a spiking neural network architecture: a case on modelling and understanding of dynamic cognitive processes," *IEEE Trans. Cogn. Dev. Syst.* (2017). <https://doi.org/10.1109/TCDS.2016.2636291>
- [29] N. K. Kasabov, *Time-space, spiking neural networks and brain-inspired artificial intelligence*. Springer, 2019.
- [30] [30] M. G. Doborjeh, N. Kasabov, and Z. G. Doborjeh, "Evolving, dynamic clustering of spatio/spectro-temporal data in 3D spiking neural network models and a case study on EEG data," (in English), *Evolving Syst.*, Article vol. 9, no. 3, pp. 195-211, 09 / 01 / 2018.
- [31] J. Siegmund et al., "Understanding understanding source code with functional magnetic resonance imaging," *Proc. 36th Int. Conf. Softw. Eng.*, pp. 378-389, 2014.
- [32] S. Lee et al., "Comparing programming language comprehension between novice and expert programmers using EEG analysis," *IEEE 16th Int. Conf. Bioinf. BIBE*, pp. 350-355, 2016.
- [33] S. Song, K. D. Miller, and L. F. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neuroscience* vol. 3, pp. 919-926., Sept 2000.
- [34] X. Hou, Y. Liu, O. Sourina, and W. Mueller-Wittig, "CogniMeter: EEG- based emotion, mental workload and stress visual monitoring," in *2015 Int. Conf. on Cyberworlds (CW)*, 2015, pp. 153-160: IEEE.